بسم الله الرحمن الرحیم

شبکه های کامپیوتری

دکتر پاکروان

گزارش تمرین سری ۲

امیرحسین رستمی ۹۶۱۰۱۶۳۵

گزارش تمرین شامل دو بخش است:

بخش اول: پاسخ دهی به سوالات پرسیده شده.

بخش دوم: پاسخ دهی به بخش MATLAB Simulation.

- بخش اول:پاسخ دهی به سوالات پرسیده شده در متن.

1-What services are provided by the data link layer for the network layer?

- Unacknowledged Connectionless
- Acknowledged Connectionless
- Acknowledged connection oriented

Note: Unacknowledged connection oriented is too meaningless and doesn't exist.

2- Why a timer is used in the transmitter?

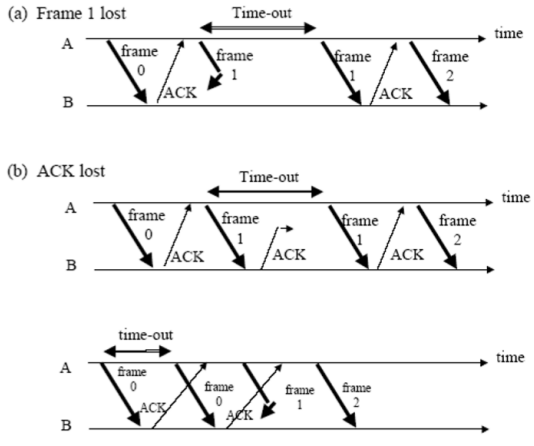
دلیل اول:

فرض کنید فرستنده packet ارسالی اش دچار مشکل شود حین مسیر و به دست گیرنده نرسد،در این صورت گیرنده اصلا Acknowledge نمی فرستند و اشتباها هم گیرنده همینطور منتظر دریافت Acknowledge می گردد و لذا سامانه وارد dead block می شود و تبادل قطع می شود لذا لازم است که از یک تایمر استفاده شود که اگر Acknowledge ای نرسید مجدد ارسال داشته باشیم تا بالاخره در گیرنده دریافت شده و Acknowledge آن نیز به دست فرستنده برسد.

دلیل دوم:

گاهی وقت ها که در فرستنده ما منتظر گرفتن Acknowledge ایم(و فرض کنید برخلاف حالت قبل بسته به گیرنده رسیده باشد) ممکن است بنا به دلایلی(از قبیل کم کیفیت بودن کانال) این Acknowledge به ما نرسد(یا نسبت به حالت معمولی دیر تر بخواهد برسد)،اگر قرار باشد ما مدام منتظر دفعات بعدی ارسال Acknowledge باشیم،وقت زیادی حروم منتظر ماندن برای رسیدن Acknowledge می گردد واین اصلا سناریوی هوشمندی در بهره برداری حداکثری از لینک نیست لذا در فرستنده از تایمری استفاده می کند که اگر مدت زمانی که منتظر Acknowledge می ماند از مقداری بیشتری شود،خودش بدون اینکه منتظر Acknowledge بماند دوباره بسته ی قبلی را بفرستد تا سرانجام Acknowledge به موقع برسد.در این شرایط بهینه تر از امکانات موجود استفاده می شود.

در زیر انواع اشکالات ذکر شده و کارایی استفاده از تایمر را ملاحظه می کنیم:(اگر تایمر استفاده نمی شد در مورد اول و دوم وارد deadBlock می شدیم).



3-What is the main draw-back of this Stop and wait protocol and in what condition it is suitable to use it?

First part:

- Transmission of packet using this protocol is very slow.

Second part: It works fine when

- If channels are noiseless.(it can be solved by using Sequence number)
- If there is no delay in the network.
- If queuing delay to is 0, which is not true in case of internet.

4-Is sequence number necessary for the transmitter's packets and how many bits are sufficient for the sequence number? What about the Ack packets?

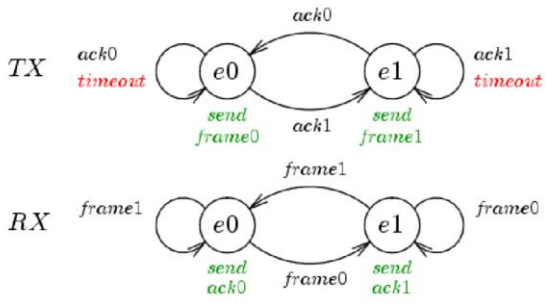
First part:

- Yes we need it because Noise always exists.
- We just need two states so it is will being solved just by using 1bit.

Second part:

- Yes we need two states just like Sequence number so it will being solved just by using 1 bit.

Look at the diagram below to figure it out.



5-What is the main draw-back of Go back N protocol? What protocol solves this problem and how?

Part one:

- A **disadvantage of Go-Back-N** is the possibility of sending frames multiple times even if they were transmitted without error.

Second part:

- Selective Repeat protocol solved this problem.

How?

Selective Repeat attempts to retransmit only those packets that are actually lost (due to errors) Receiver must be able to accept packets out of order and Since receiver must release packets to higher layer in order, the receiver must be able to buffer some packets.

Now we should have some retransmission requests, there are 2 way of doing this:

1-Implicit:

- The receiver acknowledges every good packet, packets that are not ACKed before a time-out are assumed lost or in error. Notice that this approach **must** be used to be sure that every packet is eventually received

2-Explicit:

- An explicit NAK (selective repeat) can request retransmission of just one packet, this approach can **expedite** the retransmission but is not strictly needed.

One or both approaches are used in practice.

I thought that maybe the question asked the main algorithm so I brought it:

Selective Repeat Algorithm

Sender:

- Can transmit new packets as long as their number is within W (Window Size) of all unACKed packets
- Retransmits un-ACKed packets after a timeout (Or upon a NAK if NAK is employed)
- Must buffer all packets until they are ACKed (Up to W un-ACKed packets are possible)

Receiver is in the next page.

Receiver:

- ACKs all correct packets
- Stores correct packets until they can be delivered in order to the higher layer
- Must buffer packets until they can be delivered in order i.e., until all lower numbered packets have been received. This is needed for orderly delivery of packets to the higher layer
- Up to W packets may have to be buffered (in case the first packet of a window is lost)

6-How many bits are sufficient for the sequence number and why?

It should use at least Log2(M) bits, which M is >= 2W and W is the window Size.

- بخش دوم:

نکته مهم: پیش از بررسی این دو پروتکل لازم است در ابتدا نکاتی راجع به روند گزارش ذکر کنم:
از آنجا که روابط Link Utilization وابسته به $a = \frac{T_{prop}}{T_{frame}}$ اند لذا بنده به جای جاروب روی تک T_{prop} یا T_{frame} بنده روی a جاروب می زنم و طبیعتا با این کار هم نکات تغییرات T_{frame} و هم نکات تغییرات T_{prop} در می آید،چرا که افزایش a تواما می تواند به معنای افزایش T_{prop} یا کاهش T_{frame} باشد و هم چنین کاهش a نیز تواما می تواند به معنای کاهش T_{prop} یا افزایش T_{frame} باشد.

حال در قسمت 0 به بررسی فقط تغییرات a می پردازیم،در قسمت 1 به بررسی تغییرات p_{loss} می پردازیم و طبیعتا برای داشتن نموداری شفاف از نحوه ی تغییر Link Utilization به جاروب تواما روی a به ازای مقادیر ثابت p_{loss} می پردازیم و در نهایت نیز سناریوی 1 را برای حالتی که p_{loss} ثابت و سایز پنجره تغییر می کند نیز انجام می دهیم.توجه کنید ک در هرجا که واژه ی بهره برداری را استفاده می کنیم منظورمان "بهره برداری از لینک" است.

نکته ی خیلی مهم:بنده در صورت تمرین اندکی دچار گنگی شده بودم که دقیقا چه چیزی را با چه روابط مطلوبی باید بررسی کنم و از آنجا که link utilization اطلاعات مفیدی و نیز مقایسه ی مفیدی از هر سه پروتکل به ما می دهد لذا بنده ابتدا به بررسی ریز و دقیق link utilization پرداختم و سپس در نهایت مقایسه ی دقیقی از هر سه پروتکل کرده ام اگر مشتاق به مشاهده ی نتایج اید به آخرین صفحات گزارش مراجعه کنید.

پروتکل Stop and Wait:

پیش از رسم و بررسی نمودار ها در ابتدا لازم است که روابط لازم در این بخش را بیان کنیم که برای این کار به اسلاید شماره 38 سری اسلاید سری سوم که مباحث مربوط به Datalink Layer اند مراجعه می کنیم:

Link Utilization

- Probability of frame error = p
- Stop and Wait Utilization
  - Probability of a frame requiring exactly k transmission = $p^{k-1}(1-p)$
  - Expected Number of Transmission for a frame (N_r):

$$E[N] = N_r = \sum_{k=1}^{\infty} k \times \text{Prob}(k\ Transmission) = \sum_{k=1}^{\infty} kp^{k-1}(1-p) = \frac{1}{1-p}$$

  - Utilization should be divided by N_r : $U = \frac{1-p}{1+2a}$

حال به کمک روابط ذکر شده،پیاده سازی های لازم را در متلب انجام داده و نتایج هر بخش را ذکر می کنیم.

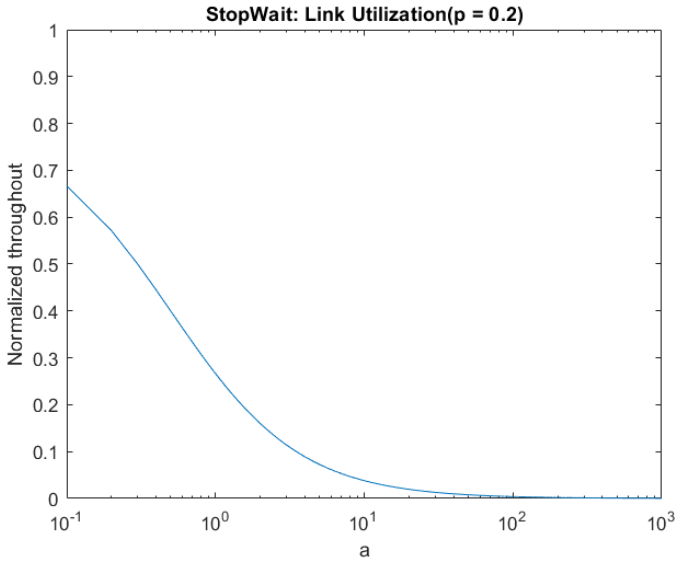
نکات نمودار ها:

۱ ـ در تک تک بخش های ۱و۲و۳ جاروب ما روی مقادیر a به شرح زیر است:

```
a = linspace(0.1,1000,number); % Tp/Tf;
```

۲ ـ در هر بخش نمودار ها در راستای محور x که همان a است به صورت لگاریتمی و در راستای محور y به صورت عادی رسم شده است.

1ـ سایز پنجره ثابت و p_{loss} ثابت و جاروی روی a:



توضیح نمودار:(می دانیم $a = \frac{T_{prop}}{T_{frame}}$)

- اثر تغییر T_{prop}(توجه در این بررسی فرض می شود که T_{frame} ثابت است)

  - افزایش: همانطورکه ملاحظه می کنید با افزایش T_{prop} مقدار a افزایش می یابد و طبیعتا انتظار ما این است که به صورت کلی با افزایش T_{prop} به علت کند شدن انتقال اطلاعات بهره برداری از کانال کاهش پیدا کند و لذا انتظار داریم که با حرکت در جهت افزایش a نمودار نزولی باشد و همونطورکه مشاهده می کنید خروجی نمودار نیز مطلب ذکر شده را تایید می کند.

  - کاهش: مشابها با کاهش T_{prop} مقدار a کاهش می یابد و طبیعتا انتظار ما این است که به صورت کلی با کاهش T_{prop} به علت سریع تر منتقل شدن اطلاعات بهره برداری از کانال افزایش پیدا کند و لذا انتظار داریم که با حرکت در جهت کاهش a نمودار صعودی باشد و همونطورکه مشاهده می کنید خروجی نمودار در تایید با مطلب ذکر شده است.

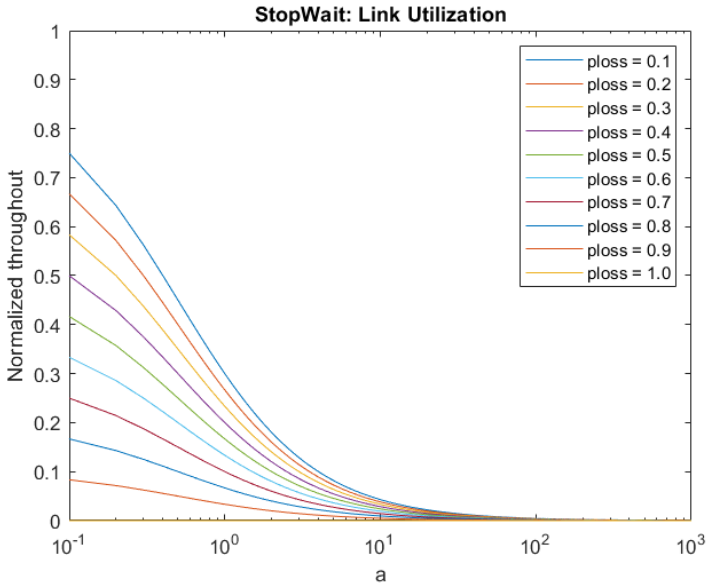
- اثر تغییر T_{frame}(توجه در این بررسی فرض می شود که T_{prop} ثابت است)

  - افزایش: همانطورکه ملاحظه می کنید با افزایش T_{frame} مقدار a کاهش پیدا می کند و لذا در نمودار به سمت چپ در حال حرکت ایم،از طرفی با توجه به معنی و مفهوم T_{frame} افزایش آن در معنا به نفع بهره برداری ما از کانال است ولذا باید با افزایش T_{frame} (T_{prop} در ثابت)روند صعودی باشد و همانطورکه در نمودار ملاحظه می کنید نیز با افزایش آن که منتها به کاهش a می شود میزان بهره برداری افزایش می یابد☺.

  - کاهش: همانطورکه ملاحظه می کنید با کاهش T_{frame} مقدار a افزایش پیدا می کند و لذا در نمودار به سمت راست در حال حرکت ایم،از طرفی با توجه به معنی و مفهوم T_{frame} کاهش آن(T_{prop} در ثابت) با توجه به نحوی عملکرد پروتکل به ضرر بهره برداری ما از کانال است و لذا باید با کاهش T_{frame} روند نزولی باشد و همانطورکه در نمودار ملاحظه می کنید نیز با کاهش آن که منتها به افزایش a می شود میزان بهره برداری کاهش می یابد☹.

نکته بسیار مهم:همانطورکه می دانید یکی از ضعف های پروتکل GoBackN این است که بدون فرآیندِ windowing اطلاعات را انتقال می دهد و لذا همواره افزایشa به ضرر ماست! لذا همانطورکه ملاحظه کردید نمودار با افزایش a اکیدا نزولی است،اما جالب است بدانید که این مشکل تا حدی در پروتکل Selective Repeat بهبود پیدا کرده است و آن هم به لطف انتقال window گونه ی اطلاعات است که باعث می شود قبل از آن که a از حدی بزرگتر شود،بهره برداری ما از کانال ثابت بماند.

2ـ سایز پنجره ثابت و جاروب روی p_{loss} :



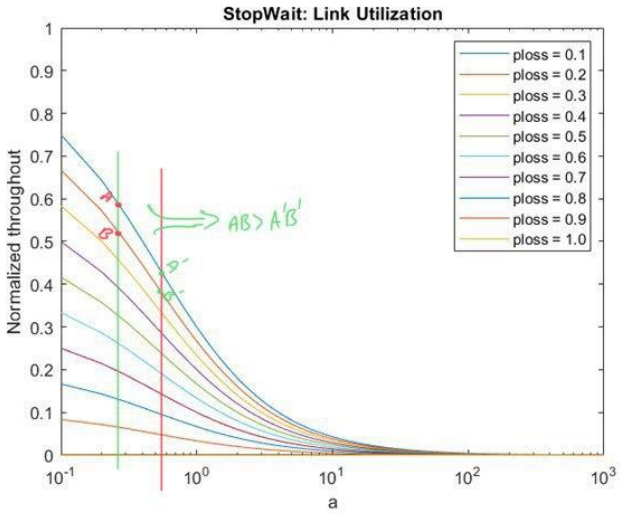
توضیح نمودار:

- اثر تغییر p_{loss}

  - کاهش: می دانیم که p_{loss} به معنای احتمال رخ داد خطاست و خب واضح است که در شرایط ثابت(ثابت بودن مقدار a و سایز پنجره) باکاهش این احتمال(وبه تبع آن احتمال بالا انتقال درست اطلاعات) میزان بهره برداری ما از کانال افزایش پیدا می کند،حال اگر به نمودار های بالا مراجعه کنیم،و در یک نقطه ی دلخواه از محور a(یعنی یک مقدار ثابت a) به صورت کاملا عمودی به بالا حرکت کنیم(یعنی در راستای افزایش بهره برداری از کانال قدم برداریم)مشاهده می کنیم که نمودارهایی که روی آن ها قدم می گذاریم از نمودار با بیشترین احتمال خطا شروع می شود و به نمودار باکمترین احتمال خطا منتها می شود.دقت کنید که در اولین قدم که روی محور a هستیم و یعنی میزان بهره برداری ما(مقدار محور y) صفر است انتظار داریم که احتمال خطا 1 باشد!(که در نمودار ها هم این چنین است و نمودار ها نیز گفته ی ما را تایید می کنند) چراکه وقتی کاملا روی بهره برداری 0 هستیم 😐 و لذا باید هرگز داده ای درست انتقال پیدا نکند و این یعنی احتمال خطا باید 1 (مقدار ماکزیمم خود) باشد.

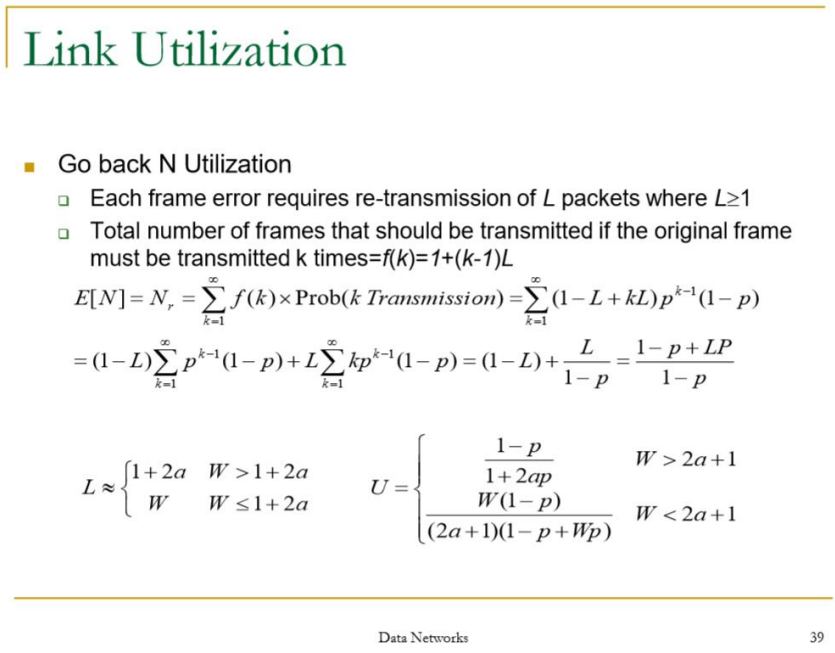
- افزایش: مشابه توضیحات قسمت افزایش،در این حالت انتظار داریم که با افزایش مقدار احتمال خطا(یعنی کاهش احتمال انتقال درست اطلاعات)میزان بهره برداری ما کاهش یابد،حال همانطور که در legend نمودار مشخص است داریم که با افزایش میزان p_{loss} ((در مقدار ثابت a) -> خط عمودی بر محور y)به سمت محور y در حال حرکت ایم یعنی بهره برداری ما در حال کاهش هست.

یک نکته جالب و مهم:همانطورکه در نمودار ها ملاحظه می کنید هرچه a افزایش یابد(یعنی وضعیت کانال خراب تر شود) بدیهی است که بهره برداری ما کاهش پیدا می کند و خب منطقی است که در چنین شرایطی میزان تغییرات بهره برداری ما به ازای تغییرات p_{loss} کمتر است نسبت به حالتی که a کم باشد،این نکته به صورت شفاف تر در نمودار زیر مشخص شده است:



پروتکل GoBackN:

پیش از رسم و بررسی نمودار ها در ابتدا لازم است که روابط لازم در این بخش را بیان کنیم که برای این کار به اسلاید شماره 39 سری اسلاید سری سوم که مباحث مربوط به Datalink Layer اند مراجعه می کنیم:



حال به کمک روابط ذکر شده،پیاده سازی های لازم را در متلب انجام داده و نتایج هر بخش را ذکر می کنیم.

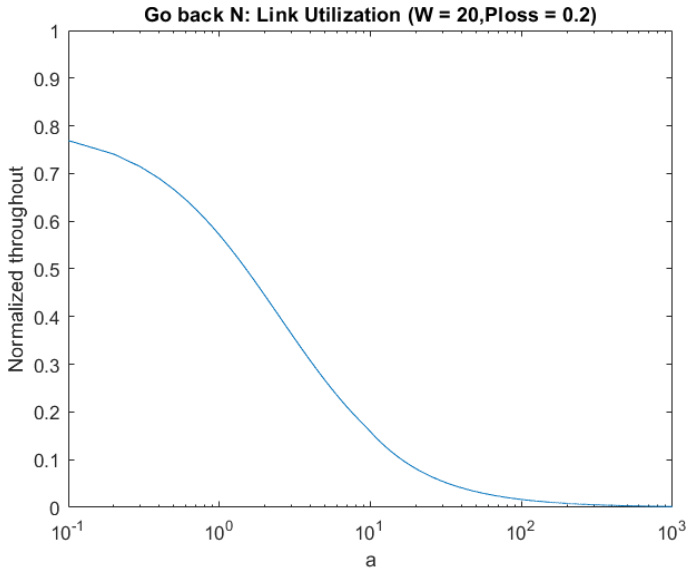
نکات نمودار ها:

۱ ـ در تک تک بخش های ۱ و۲ و۳ جاروب ما روی مقادیر a به شرح زیر است:

```
a = linspace(0.1,1000,number); % Tp/Tf;
```

۲ ـ در هر بخش نمودار ها در راستای محور x که همان a است به صورت لگاریتمی و در راستای محور y به صورت عادی رسم شده است.

1ـ سایز پنجره ثابت و p_{loss} ثابت و جاروی روی a:



توضیح نمودار:(می دانیم $a = \frac{T_{prop}}{T_{frame}}$)

- اثر تغییر T_{prop}(توجه در این بررسی فرض می شود که T_{frame} ثابت است)

  - افزایش: همانطورکه ملاحظه می کنید با افزایش T_{prop} مقدار a افزایش می یابد و طبیعتا انتظار ما این است که به صورت کلی با افزایش T_{prop} به علت کند شدن انتقال اطلاعات بهره برداری از کانال کاهش پیدا کند و لذا انتظار داریم که با حرکت در جهت افزایش a نمودار نزولی باشد و همونطورکه مشاهده می کنید خروجی نمودار نیز مطلب ذکر شده را تایید می کند.

  - کاهش: مشابها با کاهش T_{prop} مقدار a کاهش می یابد و طبیعتا انتظار ما این است که به صورت کلی با کاهش T_{prop} به علت سریع تر منتقل شدن اطلاعات بهره برداری از کانال افزایش پیدا کند و لذا انتظار داریم که با حرکت در جهت کاهش a نمودار صعودی باشد و همونطورکه مشاهده می کنید خروجی نمودار در تایید با مطلب ذکر شده است.

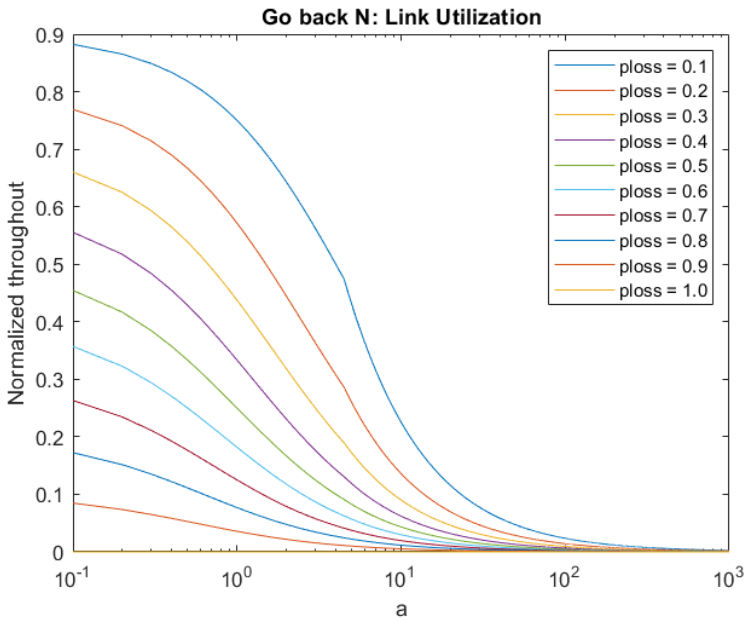
- اثر تغییر T_{frame}(توجه در این بررسی فرض می شود که T_{prop} ثابت است)

  - افزایش: همانطورکه ملاحظه می کنید با افزایش T_{frame} مقدار a کاهش پیدا می کند و لذا در نمودار به سمت چپ در حال حرکت ایم،از طرفی با توجه به معنی و مفهوم T_{frame} افزایش آن در معنا به نفع بهره برداری ما از کانال است ولذا باید با افزایش T_{frame} (در T_{prop} ثابت)روند صعودی باشد و همانطورکه در نمودار ملاحظه می کنید نیز با افزایش آن که منتها به کاهش a می شود میزان بهره برداری افزایش می یابد☺.

  - کاهش: همانطورکه ملاحظه می کنید با کاهش T_{frame} مقدار a افزایش پیدا می کند و لذا در نمودار به سمت راست در حال حرکت ایم،از طرفی با توجه به معنی و مفهوم T_{frame} کاهش آن(در T_{prop} ثابت) با توجه به نحوی عملکرد پروتکل به ضرر بهره برداری ما از کانال است و لذا باید با کاهش T_{frame} روند نزولی باشد و همانطورکه در نمودار ملاحظه می کنید نیز با کاهش آن که منتها به افزایش a می شود میزان بهره برداری کاهش می یابد☹.

نکته بسیار مهم:همانطورکه می دانید یکی از ضعف های پروتکل GoBackN این است‌که تحت هر سایز پنجره ای! همواره افزایشa به ضرر ماست!(با توجه به نحوه ی کار آن‌که کل پنجره را در صورت خطا مجدد ارسال می کند) و لذا همانطورکه ملاحظه کردید نمودار با افزایش a اکیدا نزولی است،اما جالب است بدانید که این مشکل تا حدی در پروتکل Selective Repeat بهبود پیدا کرده است.

2ـ سایز پنجره ثابت و جاروب روی p_{loss} :



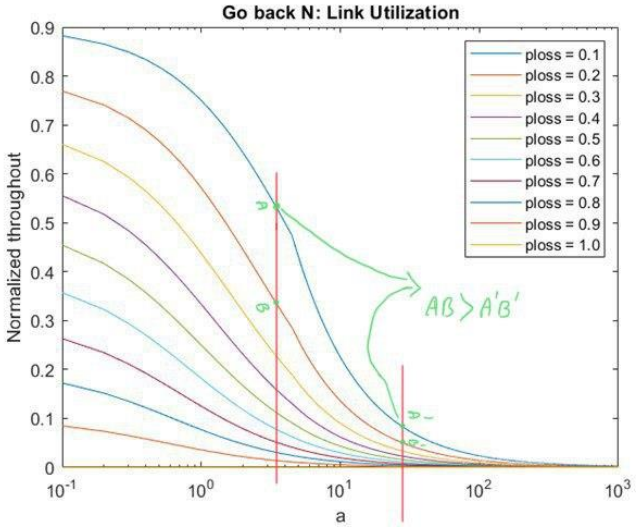
توضیح نمودار:

- اثر تغییر p_{loss}

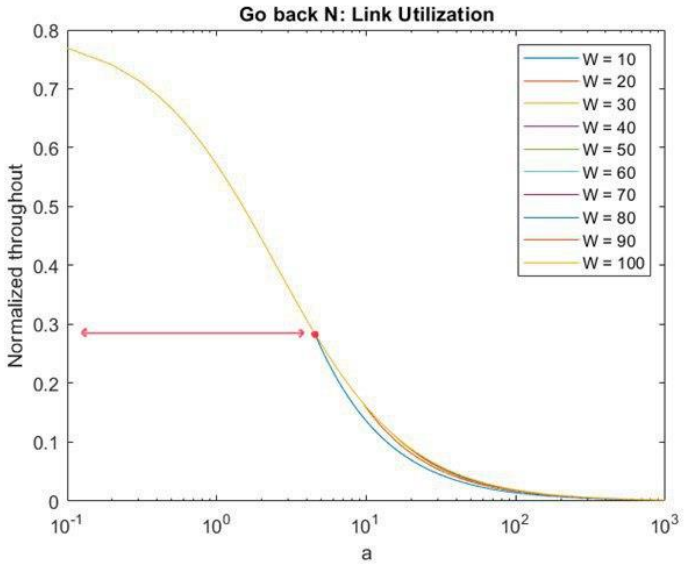
  - کاهش: می دانیم‌که p_{loss} به معنای احتمال رخ داد خطاست و خب واضح است‌که در شرایط ثابت(ثابت بودن مقدار a و سایز پنجره) باکاهش این احتمال(وبه تبع آن احتمال بالا انتقال درست اطلاعات) میزان بهره برداری ما از کانال افزایش پیدا می کند،حال اگر به نمودار های بالا مراجعه کنیم،و در یک نقطه ی دلخواه از محور a(یعنی یک مقدار ثابت a) به صورت کاملا عمودی به بالا حرکت کنیم(یعنی در راستای افزایش بهره برداری از کانال قدم برداریم)مشاهده می کنیم‌که نمودارهایی‌که روی آن ها قدم می گذاریم از نمودار با بیشترین احتمال خطا شروع می شود و به نمودار باکمترین احتمال خطا منتها می شود.دقت کنید که در اولین قدم‌که روی محور a هستیم و یعنی میزان بهره برداری ما(مقدار محور y) صفر است انتظار داریم‌که احتمال خطا 1 باشد!(که در نمودار ها هم این چنین است و نمودار ها نیز گفته ی ما را تایید می کنند) چراکه وقتی‌کاملا روی بهره برداری 0 هستیم ☹ و لذا باید هرگز داده ای درست انتقال پیدا نکند و این یعنی احتمال خطا باید 1 (مقدار ماکزیمم خود) باشد.

  - افزایش: مشابه توضیحات قسمت افزایش،در این حالت انتظار داریم‌که با افزایش مقدار احتمال خطا(یعنی‌کاهش احتمال انتقال درست اطلاعات)میزان بهره برداری ماکاهش یابد،حال همانطورکه در legend نمودار مشخص است داریم‌که با افزایش میزان p_{loss} ((در مقدار ثابت a) -> خط عمودی بر محور y)به سمت محور y در حال حرکت ایم یعنی بهره برداری ما در حال‌کاهش هست.

یک نکته جالب و مهم:همانطورکه در نمودار ها ملاحظه می کنید هرچه a افزایش یابد(یعنی وضعیت کانال خراب تر شود) بدیهی است که بهره برداری ما کاهش پیدا می کند و خب منطقی است که در چنین شرایطی میزان تغییرات بهره برداری ما به ازای تغییرات p_{loss} کمتر است نسبت به حالتی که a کم باشد،این نکته به صورت شفاف تر در نمودار زیر مشخص شده است:



3ـ مقدار p_{loss} ثابت و جاروب روی سایز پنجره :



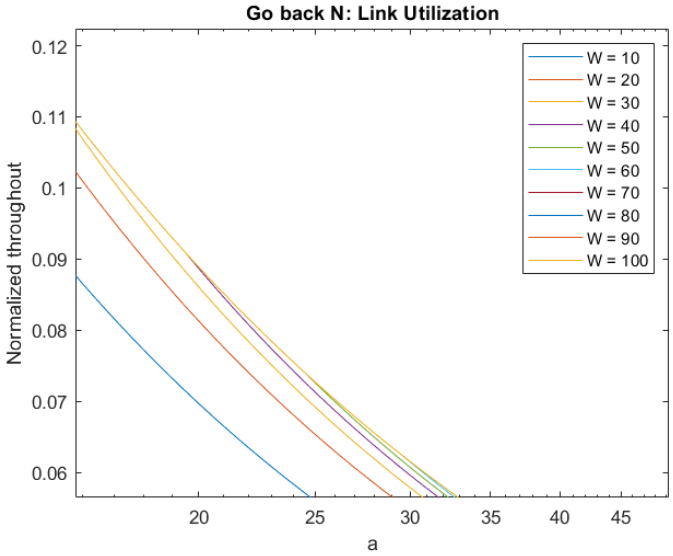
توضیح نمودار:

- اثرتغییر W

  - تغییرِ بی تاثیر!:با توجه به محاسبات موجود در اسلاید 39 (که در بالا آورده شده بود) مقدار W در حالتی که داشته باشیم که $W > 2a + 1$،داریم که کاملا مسیر در زمانی که Ack به مبدا رسیده است(از سمت مقصد) پر شده است لذا داریم که بهره برداری ما در این حالت مستقل از W است و به a و p_{loss} فقط بستگی خواهد داشت لذا انتظار داریم که با افزایش W در ابتدای نمودار ها که a آنقدر زیاد نیست که نامساوی را نقض کند نمودار های W های مختلف روی همدیگر قرار بگیرد،حال همانطور که ملاحظه می کنید در نمودار ضمیمه شده هم داریم که تا هنگامی که نمودار آبی جدا نشده است(قطعه بخشی که با ← → نشان داده شده است دقیقا بخشی است که همه ی این نمودار ها روی یکدیگر قرار گرفته اند).

حال در تک مورد بررسی های بعدی فرض می کنیم که نامساوی $W < 2a + 1$ برقرار باشد(در بخشی از نمودار بالا که نمودار ها شروع به جدا شدن از هم می کنن zoom می کنیم)،ادامه ی موارد بررسی شده را در صفحه ی بعد دنبال کنید.

- تغییرِ با تاثیر: می دانیم که افزایش سایز W در حالتی که موارد دیگر ثابت باشد به خودی خود باعث می شود که میزان بهره برداری ما از لینک افزایش یابد(بدیهی است!) لذا انتظار داریم که نمودار های با W بیشتر در a های ثابت بالا تر از نمودار های با W کمتر قرار داشته باشند،حال اگر اندکی در بخش بعد از نقطه ی قرمز نمودار بالا زوم کنیم به نمودار زیر می رسیم که مطلب بیان شده واضحا در آن دیده می شود.

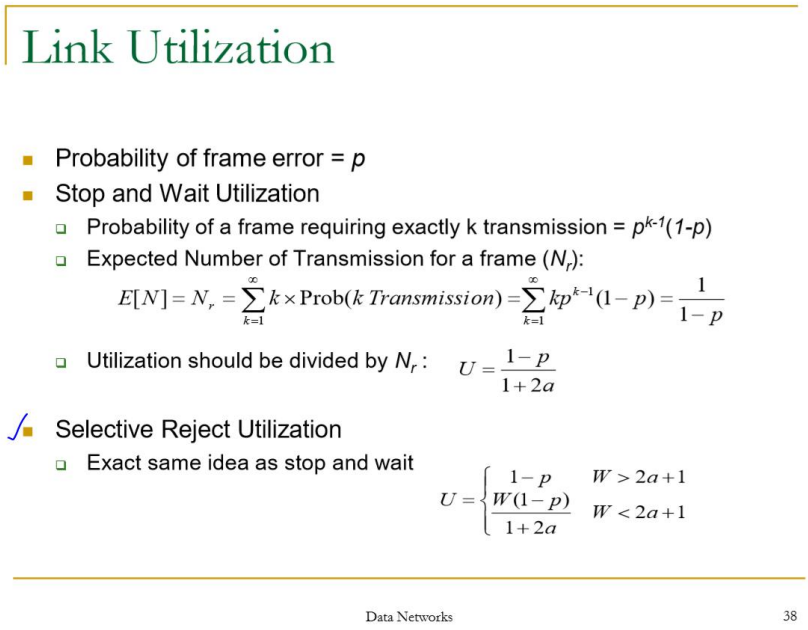


همانطور که ملاحظه می کنید نمودار با W=10 که به رنگ آبی است در پایین تر از همه قرار دارد و نمودار با W=20 که به رنگ نارنجی است بالای نمودار W=10 و زیر نمودار با W = 30 است.

نکته مهم: اگر بار دیگر به نامساوی $W > 2a + 1$ توجه کنید می ببنیم که به ازای افزایش W،آن a ای که از آن به بعد این نامساوی نقض می شود،نیز افزایش می باید لذا انتظار داریم که نمودار ها به ترتیب مقدار W خود از نمودار با بیشترین W جدا بشوند و همانطور که ملاحظه می کنید همین اتفاق نیز رخ می دهد و اولین نموداری که جدا می شود نمودار با W = 10 (که کمترین مقدار W بین نمودار ها است) است و به ترتیب بعد از آن نمودار با W های بیشتر جدا می شوند.

پروتکل Selective Repeat:

پیش از رسم و بررسی نمودار ها در ابتدا لازم است که روابط لازم در این بخش را بیان کنیم که برای این کار به اسلاید شماره 38 سری اسلاید سری سوم که مباحث مربوط به Datalink Layer اند مراجعه می کنیم:



Link Utilization

- Probability of frame error = p
- Stop and Wait Utilization
  - Probability of a frame requiring exactly k transmission = $p^{k-1}(1-p)$
  - Expected Number of Transmission for a frame (N_r):

$$E[N] = N_r = \sum_{k=1}^{\infty} k \times \text{Prob}(k \text{ Transmission}) = \sum_{k=1}^{\infty} kp^{k-1}(1-p) = \frac{1}{1-p}$$

  - Utilization should be divided by N_r : $U = \frac{1-p}{1+2a}$
- Selective Reject Utilization
  - Exact same idea as stop and wait

$$U = \begin{cases} 1-p & W > 2a+1 \\ \frac{W(1-p)}{1+2a} & W < 2a+1 \end{cases}$$

Data Networks 38

حال به کمک روابط ذکر شده،پیاده سازی های لازم را در متلب انجام داده و نتایج هر بخش را ذکر می کنیم.

نکات نمودار ها:

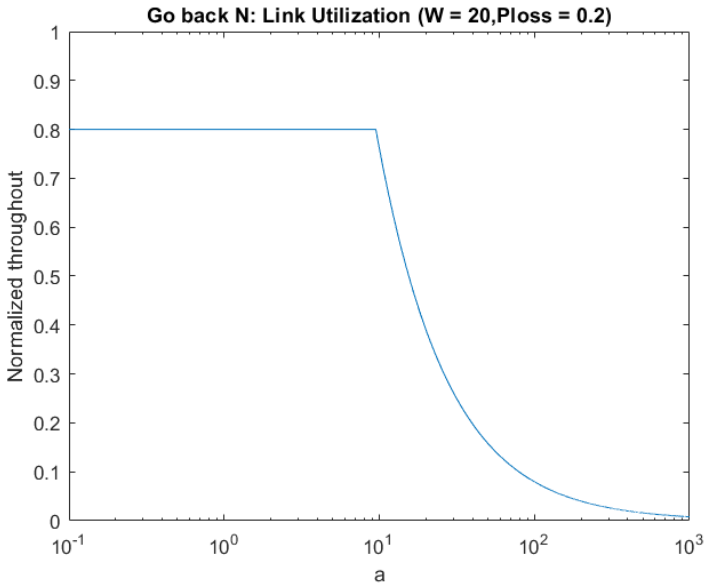
۱ ـ در تک تک بخش های ۱ و۲ و۳ جاروب ما روی مقادیر a به شرح زیر است:

```
a = linspace(0.1,1000,number); % Tp/Tf;
```

۲ ـ در هر بخش نمودار ها در راستای محور x که همان a است به صورت لگاریتمی و در راستای محور y به صورت عادی رسم شده است.

(توجه کنید که میزان جاروب روی a را مشابه پروتکل GoBackN در نظر گرفتیم تا در نهایت نیز نمودار های دو پروتکل را با یکدیگر مقایسه کنیم).

1ـ سایز پنجره ثابت و p_{loss} ثابت و جاروی روی a:



توضیح نمودار:(می دانیم $a = \frac{T_{prop}}{T_{frame}}$)

- تغییر T_{prop} و T_{frame} بی تاثیر!

نکته بسیار مهم: همانطورکه می دانید یکی از نقاط قوت پروتکل Selective Repeat نسبت به GoBackN این است که دیگر ما برخلاف پروتکل GoBackN در هر بار رخ داد خطا،همه ی packet های پنجره را برنمی گردانیم! لذا خیلی منطقی است که تا زمانی که شرط $W > 2a + 1$ برقرار باشد داشته باشیم که تغییر a روی بهره برداری ما تاثیر نداشته باشد و در این حالت بهره برداری تنها تابع p_{loss} خواهد بود لذا همانطورکه ملاحظه می کنید در نمودربالا،نمودار تا نقطه ای، کاملا بهره برداری ثابت دارد(چون نامساوی هنوز برقرار و p_{loss} ثابت است لذا بهره برداری نیز ثابت خواهد بود.

حال در تک مورد بررسی های بعدی فرض می کنیم که نامساوی $W < 2a + 1$ برقرار باشد(بخشی از نمودار بالا که نمودار ناگهان می شکند).

- اثر تغییر T_{prop}(توجه در این بررسی فرض می شود که T_{frame} ثابت است)

- افزایش: همانطورکه ملاحظه می کنید با افزایش T_{prop} مقدار a افزایش می یابد و طبیعتا انتظار ما این است که به صورت کلی با افزایش T_{prop} به علت کند شدن انتقال اطلاعات بهره برداری از کانال کاهش پیدا کند و لذا انتظار داریم که با حرکت در جهت افزایش a نمودار نزولی باشد و همونطورکه مشاهده می کنید خروجی نمودار نیز مطلب ذکر شده را تایید می کند.

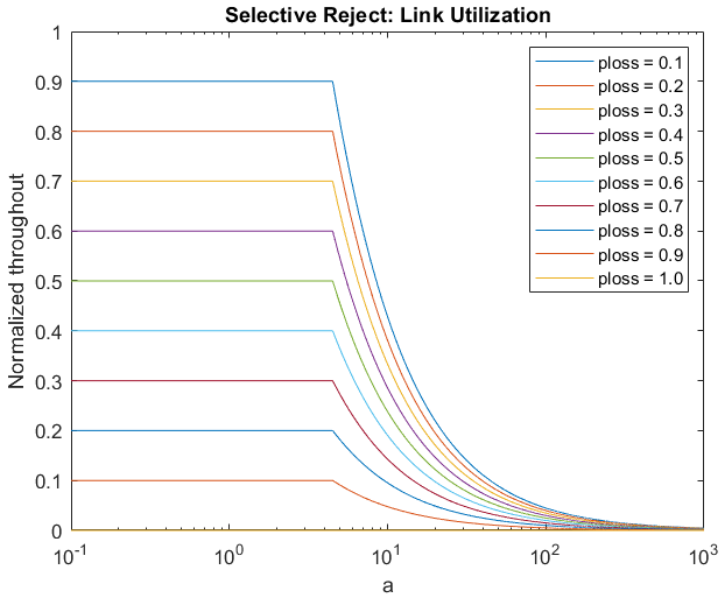
- کاهش: مشابها با کاهش T_{prop} مقدار a کاهش می یابد و طبیعتا انتظار ما این است که به صورت کلی با کاهش T_{prop} به علت سریع تر منتقل شدن اطلاعات بهره برداری از کانال افزایش پیدا کند و لذا انتظار داریم که با حرکت در جهت کاهش a نمودار صعودی باشد و همونطورکه مشاهده می کنید خروجی نمودار در تایید با مطلب ذکر شده است.

- اثر تغییر T_{frame}(توجه در این بررسی فرض می شود که T_{prop} ثابت است)

  - افزایش: همانطورکه ملاحظه می کنید با افزایش T_{frame} مقدار a کاهش پیدا می کند و لذا در نمودار به سمت چپ در حال حرکت ایم،از طرفی با توجه به معنی و مفهوم T_{frame} افزایش آن در معنا به نفع بهره برداری ما از کانال است ولذا باید با افزایش T_{frame} (T_{prop} در ثابت)روند صعودی باشد و همانطورکه در نمودار ملاحظه می کنید نیز با افزایش آن که منتها به کاهش a می شود میزان بهره برداری افزایش می یابد☺.

  - کاهش: همانطورکه ملاحظه می کنید با کاهش T_{frame} مقدار a افزایش پیدا می کند و لذا در نمودار به سمت راست در حال حرکت ایم،از طرفی با توجه به معنی و مفهوم T_{frame} کاهش آن(T_{prop} در ثابت) با توجه به نحوی عملکرد پروتکل به ضرر بهره برداری ما از کانال است و لذا باید با کاهش T_{frame} روند نزولی باشد و همانطورکه در نمودار ملاحظه می کنید نیز با کاهش آن که منتها به افزایش a می شود میزان بهره برداری کاهش می یابد☹.

2ـ سایز پنجره ثابت و جاروب روی مقدار p_{loss}:



توضیح نمودار:

- اثر تغییر p_{loss}

  - کاهش: می دانیم که p_{loss} به معنای احتمال رخ داد خطاست و خب واضح است که در شرایط ثابت(ثابت بودن مقدار a و سایز پنجره) باکاهش این احتمال(وبه تبع آن احتمال بالا انتقال درست اطلاعات) میزان بهره برداری ما از کانال افزایش پیدا می کند،حال اگر به نمودار های بالا مراجعه کنیم،و در یک نقطه ی دلخواه از محور a(یعنی یک مقدار ثابت a) به صورت کاملا عمودی به بالا حرکت کنیم(یعنی در راستای افزایش بهره برداری از کانال قدم برداریم)مشاهده می کنیم که نمودارهایی که روی آن ها قدم می گذاریم از نمودار با بیشترین احتمال خطا شروع می شود و به نمودار باکمترین احتمال خطا منتها می شود.دقت کنید که در اولین قدم که روی محور a هستیم و یعنی میزان بهره برداری ما(مقدار محور y) صفر است انتظار داریم که احتمال خطا 1 باشد!(که در نمودار ها هم این چنین است و نمودار ها نیز گفته ی ما را تایید می کنند) چراکه وقتی کاملا روی بهره برداری 0 هستیم 😐 و لذا باید هرگز داده ای درست انتقال پیدا نکند و این یعنی احتمال خطا باید 1 (مقدار ماکزیمم خود) باشد.

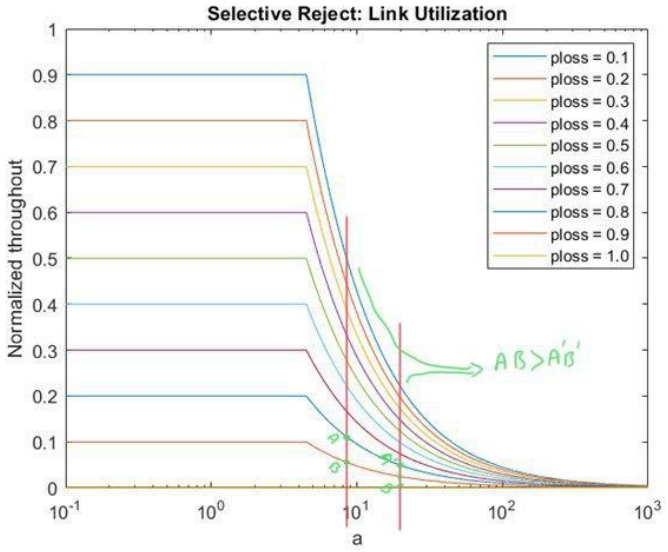
- افزایش: مشابه توضیحات قسمت افزایش،در این حالت انتظار داریم که با افزایش مقدار احتمال خطا(یعنی کاهش احتمال انتقال درست اطلاعات)میزان بهره برداری ما کاهش یابد،حال همانطور که در legend نمودار مشخص است داریم که با افزایش میزان p_{loss} ((در مقدار ثابت a) -> خط عمودی بر محور y)به سمت محور y در حال حرکت ایم یعنی بهره برداری ما در حال کاهش هست.

دو نکته جالب و مهم:

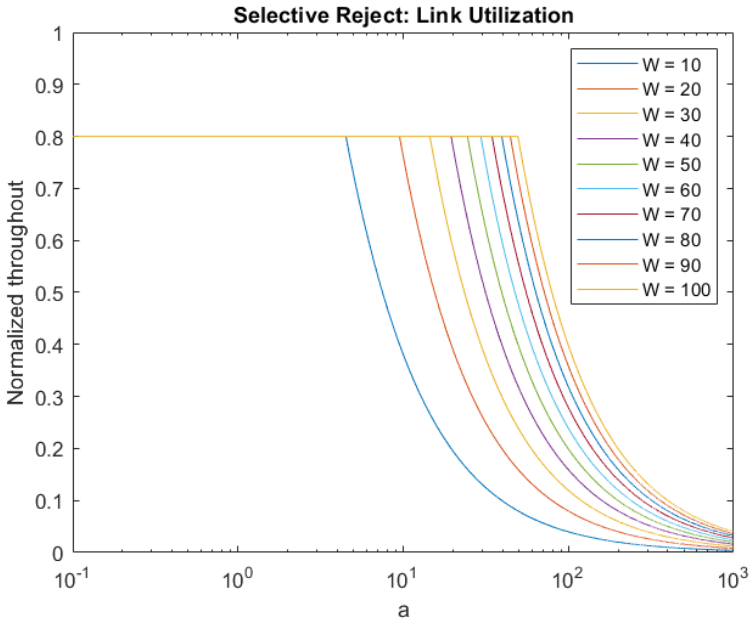
1- ناحیه ثابت اولیه!: توجه کنید که در ابتدای نمودار ها مشاهده می کنیم که با تغییر a بهره برداری تغییر نمی کند و این به این دلیل است که همانطور که می دانید یکی از نقاط قوت پروتکل Selective Repeat نسبت به GoBackN این است که دیگر ما برخلاف پروتکل GoBackN در هر بار رخ داد خطا،همه ی packet های پنجره را برنمی گردانیم! لذا خیلی منطقی است که تا زمانی که شرط $W > 2a + 1$ برقرار باشد داشته باشیم که تغییر a روی بهره برداری ما تاثیر نداشته باشد و در این حالت بهره برداری تنها تابع p_{loss} خواهد بود لذا همانطور که ملاحظه می کنید در نمودارهالا،در هر تک نمودار،این مقدار بهره برداری برابر $1 - p_{loss}$ است و لذا وقتی روی هر نمودار هستیم این مقدار ثابت است و نمودار های مختلف هم در شروع با فاصله **دقیقا برابر 0.1** از یکدیگر قرار دارند(یعنی فاصله ی بین بهره برداری نمودار ها پیش از رسیدن به نقطه ای که نامساوی ذکر شده نابرقرار شود،مقدار ثابت 0.1 است).

نکته ی زیر با فرض قرار گرفتن در بخش سمت راستِ نقطه ای که نمودار می شکند برقرار است.

2- همانطورکه در نمودار ها ملاحظه می کنید هرچه a افزایش یابد(یعنی وضعیت کانال خراب تر شود) بدیهی است که بهره برداری ما کاهش پیدا می کند و خب منطقی است که در چنین شرایطی میزان تغییرات بهره برداری ما به ازای تغییرات p_{loss} کمتر است نسبتبه حالتی که a کم باشد،این نکته به صورت شفاف تر در نمودار زیر مشخص شده است:



3ـ مقدار p_{loss} ثابت و جاروب روی سایز پنجره:



توضیح نمودار:

- اثرتغییر W

  - تغییرِ بی تاثیر!:با توجه به محاسبات موجود در اسلاید 38 (که در بالا آورده شده بود) مقدار W در حالتی که داشته باشیم که $W > 2a + 1$،داریم که کاملا مسیر در زمانی که Ack به مبدا رسیده است(از سمت مقصد) پر شده است لذا داریم که بهره برداری ما در این حالت مستقل از W است و در پروتکل Selective Repeat برخلاف پروتکل GoBackN که به a و p_{loss} بستگی داشت فقط و فقط به فقط p_{loss} بستگی خواهد داشت لذا انتظار داریم که با افزایش W در ابتدای نمودار ها که a آنقدر زیاد نیست که نامساوی را نقض کند **نمودار های W های مختلف روی همدیگر قرار بگیرد و مقدار ثابتِ برابر داشته باشند چون p_{loss} در این بررسی ثابت است و همانطور که ملاحظه می کنید نمودار درج شده در بالا نیز این موضوع را تایید می کند.**

حال در تک مورد بررسی های بعدی فرض می کنیم که نامساوی $W < 2a + 1$ برقرار باشد(یعنی بخشی از نمودارِ بالا که نمودار ها می شکنند)،ادامه ی موارد بررسی شده را در صفحه ی بعد دنبال کنید.

- تغییرِ با تاثیر: می دانیم که افزایش سایز W در حالتی که موارد دیگر ثابت باشد به خودی خود باعث می شود که میزان بهره برداری ما از لینک افزایش یابد(بدیهی است!) لذا انتظار داریم که نمودار های با W بیشتر در a های ثابت بالا تر از نمودار های با W کمتر قرار داشته باشند و همانطور که در نمودار ضمیمه شده ملاحظه می کنید داریم که نموداربا W=10 که به رنگ آبی است در پایین تر از همه قرار دارد و نمودار با W=20 که به رنگ نارنجی است بالای نمودار W=10 و زیر نمودار با W = 30 است.

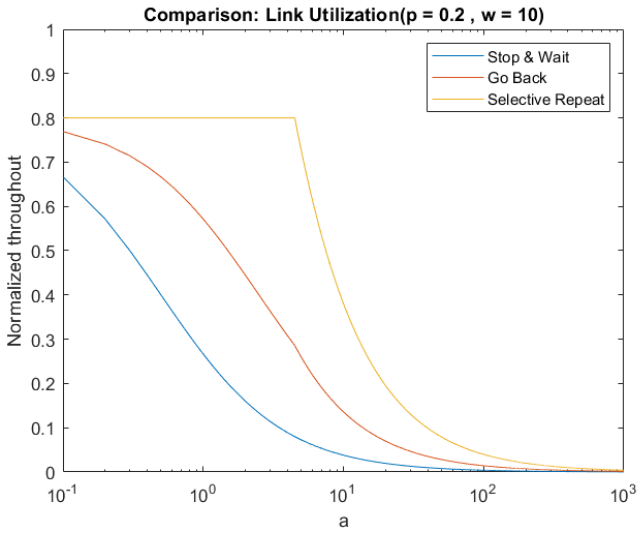
  نکته مهم: اگر بار دیگر به نامساوی $W > 2a + 1$ توجه کنید می ببنیم که به ازای افزایش W،آن a ای که از آن به بعد این نامساوی نقض می شود،نیز افزایش می باید لذا انتظار داریم که نمودار ها به ترتیب مقدار W خود از نمودار با بیشترین W جدا بشوند و همانطور که ملاحظه می کنید همین اتفاق نیز رخ می دهد و اولین نموداری که جدا می شود نمودار با W = 10 (که کمترین مقدار W بین نمودار ها است) است و به ترتیب بعد از آن نمودار با W های بیشتر جدا می شوند.

مقایسه ی پروتکل ها:

ـ (بهره برداری از کانال)

از آنجا که پروتکل Stop and Wait از windowing استفاده نمی کند لذا بدیهی است که این پروتکل در بهره برداری از لینک از دو پروتکل دیگر ضعیف تر خواهد بود و همانطورکه می دانیم پروتکل Selective Repeat از پروتکل GoBackN در بهره برداری از لینک بهتر عمل می کند(چون در صورت رخ داد خطا،فقط packet دچار اشکال را مجدد ارسال می کند اما در GoBackN کل پنجره! مجدد ارسال می گردد)و این نکته نیز در نمودار های آن ها کاملا برقرار است بدین طریق که اگر هر کدام از بخش های این دو پروتکل رو اگر در کنار هم بررسی کنید شاهد این موضوع خواهید بود که

1- در هنگامی که W و p_{loss} ثابت اند و فقط a تغییر می کند،نمودار پروتکل Selective Repeat بالاتر از نمودار پروتکل GoBackN است.(نمودار زیر را مشاهده کنید).



2- در a ثابت و تغییر روی W, p_{loss} نیز بهره برداری از لینک در پروتکل Selective Repeat از GoBackN بیشتر و هر دوی این دو پروتکل از روش Stop & Wait بهتر اند.(کافی است به نمودار های مستقل هر کدام در بالا مراجعه کنید).

ـ (Efficiency):(توجه کنید که ما خطای ارسال p را مخالف 0 گرفتیم در این بررسی)

خب اینکه در هر این حالت نیز پروتکل selective Repeat از دو پروتکل Stop and wait و GoBackN بهتر است که کاملا مبرهن است چراکه از این جهت Selective Repeat نسبت به Stop and Wait بهتر است چون که در صورت رخ داد خطا فقط تک packet مشکل دار را اصلاح می کند اما GoBackN مدام پنجره ای که ممکن است جز یکی همه اش درست بوده باشد را هی مدام ارسال کند و این خود نرخ efficiency اش را پایین می آورد و لذا نرخ ارسال درست ما شفافا در روش selective Repeat نسبت به GoBackN بیشتر است. (طبیعتا هرچه نرخ ارسال درست بالاتر باشد efficiency نیز بالاتر خواهد بود).حال کافی است GoBackN و Stop & Wait را مقایسه کنیم:در این حالت نمی توان با قطعیت گفت کدام یک نسبت به دیگر برتری دارند! و این برتری به مقدار احتمال خطای ارسال packet و سایز پنجره (w) دارد بدین گونه که:

1ـ وقتی خطای کانال خیلی زیاد است! احتمال رخ داد خطا در بسته ای از ارسال ها بیشتر از تک ارسال است،مثلا فرض کنید ploss = 0.8 حال اگر طول پنجره ی ما w باشد(w تا از packet ها را داشته باشیم)،می دانیم که ارسال مجدد انجام می شود هنگامی که حداقل یک بسته دچار اشکال شود:

$$P_{atLeastOneError} = 1 - (1 - p_{loss})^w$$

حال اگر توجه کنید،اگر ploss زیاد باشد،احتمال رخ داد حداقل یک خطا در ارسال بسته ی i تایی زیاد می شود و لذا اگر ploss و i هر دو زیاد شوند به حالتی می رسیم که احتمال حداقل یک رخ داد خطا در ارسال بسته آنقدر زیاد می گردد که جدا جدا ارسال اطلاعات از احتمال صحت ارسال بیشتری برخوردار بود و لذا در مواقعی که احتمال خطا بالاست روش Stop and Wait نسبت به روش GoBackN از efficiency بیشتری بهره مند است چراکه چون در روش GoBackN کل بسته مجدد ارسال می گشت لذا برای یک ارسال سالم در حالتی که خطای ارسال دیتا زیاد است تعداد زیاااادی ارسال که احتمال خطای زیاد داشت بایست ارسال می شد لذا در نهایت efficiency کمتری خواهد داشت.در این **حالت افزایش سایز پنجره نرخ Efficiency را بیشتر پایین** می آورد(کافی است به اثر W در فرمول سبز بالا توجه کنید).

2ـ وقتی خطای کانال کم است! احتمال رخ داد خطا در ارسال window طور کم است و نسبت به حالتی که w بار یک عدد یک عدد ارسال کنیم(Stop and Wait)،پروتکل GoBackN به صرفه تر است و با اندکی تفکر مشخص است که در این حالت efficiency عه پروتکل GoBackN بیشتر از Stop and Wait است.

ـ (time Delay):

بنده در این بررسی فرض کردم که time delay به معنای میزان معطل شدن پروتکل است(تلف شدن وقت!) و طبق این تعریف مشخص است که میزان linkUtilization با time Delay رابطه ی عکس خواهد داشت و لذا همه ی صحبت هایی که برای linkUtilization کردیم به صورت وارون برای time Delay برقرار خواهد بود و از آنجا که طبق صحبت ها و بررسی شده های انجام شده در صفحات قبل داریم میزان بهره برداری از لینک در Selective Repeat نسبت به GoBackN (در W,P ثابت) بیشتر است و هر دوی این ها از Stop and wait در بهره برداری از لینک برتر اند لذا مقایسه ی time delay وارون این سناریو خواهد بود و داریم که time Delay در پروتکل Stop and wait از time Delay در پروتکل GoBackN بیشتر بوده و timeDelay در پروتکل GoBackN نیز از timeDelay در پروتکل Selective Repeat بیشتر است.